Day 10: Practical Social Media Data Mining

Kenneth Benoit and Slava Mikhaylov

Introduction to Data Science and Big Data Analytics

26 August 2016

Day 10 Outline

Social Media Data

Accessing social media APIs

"Web scraping"

Social Media Data

Why social media data?

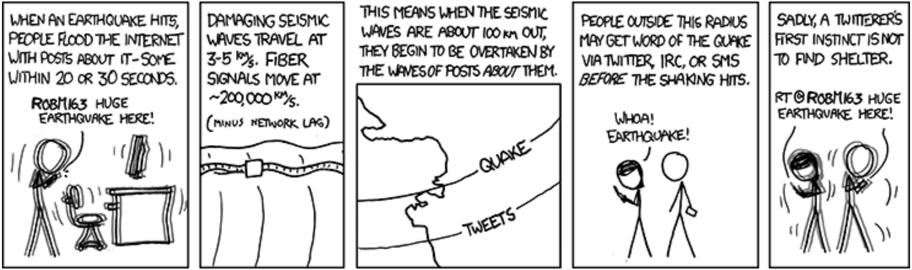
- Volume and coverage
- Twitter: 316 million monthly active users, 500m tweets per day [1]
- Facebook: 968 million daily active users on average for June 2015, 1.49 billion monthly active users as of June 30, 2015 [2]
- Real time — new data is available (somewhat) publicly immediately on current events
- Metadata — geographic location, user device, profile, timestamp and other metadata is accessible.

Appeal of Social Media data

- Good case for machine learning and data mining — lots of data, lots of metadata
- Many-to-many *broadcast* text corpus
- Social network analysis: a graph of social connections

Network data structure of social media

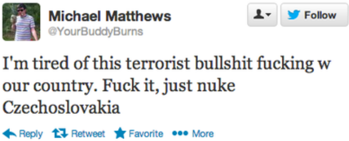
- Broadcast
  - simplex (e.g. radio, semaphore, smoke signal)
  - duplex (e.g. round-table meeting)
- Point-to-point: sender specifies receivers
- Social media allow many of these different forms of communication
- Twitter in particular is a completely new model of communication (social or news?)
- Every user is a sensor, receiver, and broadcaster — a distributed sensor network (Crooks et al 2012)


WHEN AN EARTHQUAKE HITS, PEOPLE FLOOD THE INTERNET WITH POSTS ABOUT IT–SOME WITHIN 20 OR 30 SECONDS.
ROBM163 HUGE EARTHQUAKE HERE!
DAMAGING SEISMIC WAVES TRAVEL AT 3-5 KM/S. FIBER SIGNALS MOVE AT ~200,000 KM/S.
(MINUS NETWORK LAG)
THIS MEANS WHEN THE SEISMIC WAVES ARE ABOUT 100 KM OUT, THEY BEGIN TO BE OVERTAKEN BY THE WAVES OF POSTS *ABOUT* THEM.
QUAKE
TWEETS
PEOPLE OUTSIDE THIS RADIUS MAY GET WORD OF THE QUAKE VIA TWITTER, IRC, OR SMS *BEFORE* THE SHAKING HITS.
WHOA! EARTHQUAKE!
SADLY, A TWITTERER'S FIRST INSTINCT IS NOT TO FIND SHELTER.
RT @ROBM163 HUGE EARTHQUAKE HERE!

Possible downsides

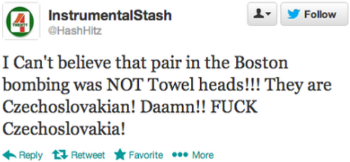
- Legal and ethical concerns
  - twitter is public, facebook private – see `https://twitter.com/tos?lang=en`
  - legal issues need to catch up with the technology
  - Are EULAs (End-User License Agreement) too complex to allow ‘informed consent’?
- Sampling issues and many new methodological headaches: homographs, people tweet about interesting events
- Biased sample (Barbera and Rivero 2013)
- commercial interfaces are brittle and opaque
- A lot of the content is moronic

Example: Twittdiots



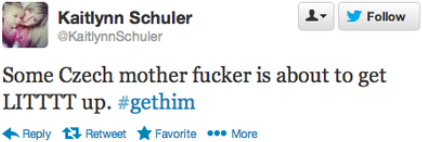
Michael Matthews
@YourBuddyBurns

I'm tired of this terrorist bullshit fucking w our country. Fuck it, just nuke Czechoslovakia



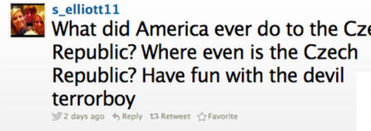
InstrumentalStash
@HashHitz

I Can't believe that pair in the Boston bombing was NOT Towel heads!!! They are Czechoslovakian! Daamn!! FUCK Czechoslovakia!

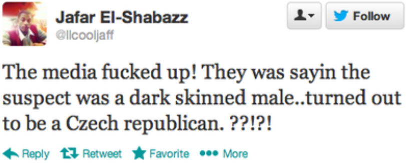


Kaitlynn Schuler
@KaitlynnSchuler

Some Czech mother fucker is about to get LITTTT up. #gethim



s_elliott11
What did America ever do to the Czech Republic? Where even is the Czech Republic? Have fun with the devil terrorboy



Jafar El-Shabazz
@llcooljaff

The media fucked up! They was sayin the suspect was a dark skinned male..turned out to be a Czech republican. ??!?!

Other twitter challenges

- Large amounts of data
  - storage problems
  - analysis problems
- Language is informal and often non-textual (emoticons, links, images) - and slang, txtspk, emoticons :-(
- lots of fake users
- A lot of the content is non-message oriented e.g. `http://twitter.com/search?q=%23JamesCallSam`

Example applications

- Tracking disease through google search terms and social media (Lampos et al 2010)
  - Locate tweets in urban centres
  - Uses a Porter stemmer and stopwords
  - Uses regression to learn which words are associated with flu outbreaks: from 1560 to 97 'markers'
  - Use this association to observe current outbreaks

Example applications

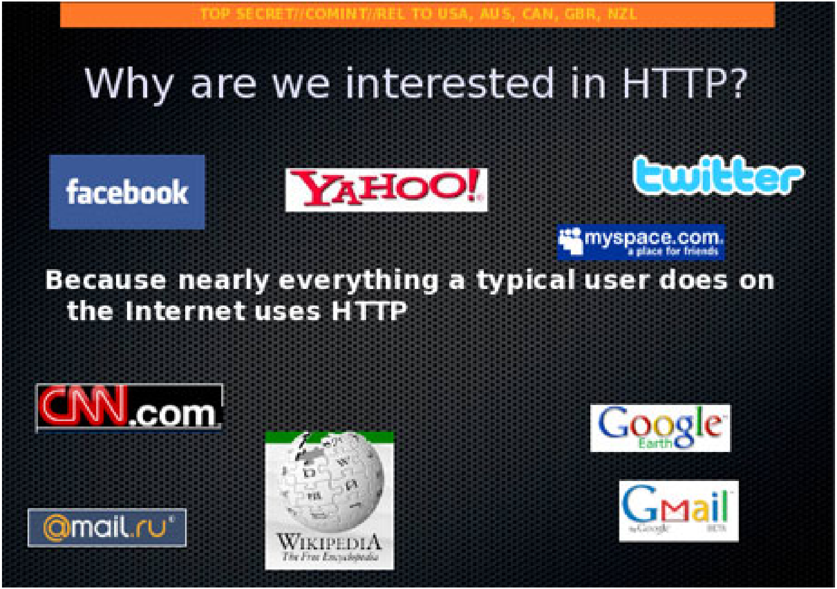
- Predicting election outcomes or polls
- Sentiment: particularly for financial or corporate interests
- (Vasileios Lampos: www.lampos.net)
- Government security/intelligence
- Social network analysis: a graph of social connections
- Nulty et al (2015) study of EP 2014

Social Media Data access

How can we access this data?

- API: Application Programming Interface — a way for two pieces of software to talk to each other
- Twitter, facebook, google — all expose public web services
- Your software can receive (and also send) data automatically through these services
- Data is sent by `http` — the same way your browser does it
- Most services have helping code (known as a wrapper) to construct http requests
- both the wrapper and the service itself are called APIs
- http service also sometimes known as REST (REpresentational State Transfer)

HyperText Transfer Protocol



TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

Why are we interested in HTTP?

Because nearly everything a typical user does on the Internet uses HTTP

facebook
YAHOO!
twitter
myspace.com a place for friends
CNN.com
Google Earth
WIKIPEDIA The Free Encyclopedia
Gmail by Google BETA
@mail.ru

Anatomy of a http request

```
https://api.twitter.com/1.1/search/tweets.json?
q=Nick+Clegg%21&since_id=24012619984051000&max_id=250126199
```

`Nick Clegg! becomes Nick+Clegg%21`

- Parameters to the API are encoded in the URL
- you must encode requests — spaces and non ASCII characters are replaced

cURL and wget

- It's not usually necessary to construct these kind of requests yourself
- R, Python, and other programming languages have libraries to make it easier
- Usually you will need cURL installed to access an API, wget for downloading a website
- The documentation for the API will describe the parameters that are available.

Available social media APIs

- Wikipedia: mediawiki
- Google
    - google plus
    - blogger
- reddit
- foursqure
- facebook
- twitter: REST, Streaming, firehose, commercial

The twitter APIs: REST

- This is the most comprehensive API
- Returns a sample of historical data from the last 8–10 days.
- Stateless: you send a command and receive a result.
- http GET requests return information
- http POST requests upload or alter information (e.g. twitterbots)
- The manual: `https://dev.twitter.com/docs/api/1.1`
- R package : twitteR

The twitter APIs: Streaming

- Connect to the twitter server and collect tweets as they fly by.
- The manual: `https://dev.twitter.com/docs/streaming-apis/streams/public`
- R package: streamR

Authentication

- Username and Password
- Oauth (ROauth): share a key without sharing a username and password
- IP address limitations
- Rate limitations
- Per-user and per-application

Other options

- The firehose: work with twitter
- Commercial options: GNIP and Datasift

The Output: JSON and XML

- XML: eXtensible Markup Language: encodes documents in a form that is both human-readable and machine readable
- JSON : JavaScript Object Notation
- If you have a choice, you probably want JSON
- JSON uses key:value pairs, XML uses trees
- JSON is easily read into a programming language
- Sometimes known as serialization formats

And finally... the text.

- Full of spam, bots, unicode, and gibberish
- Homographs are major problem, e.g. Clegg, Cameron, Miliband
- Lots of retweets
- Only 1% show location

Twitter uses: Exploiting the meta-data (non-textual)

- location
- time
- username
- user descriptions
- networks of followers
- retweets of followers and texts

Connecting through R

R packages

- Twitter: twitteR for REST, streamR for Streaming
- Facebook: Rfacebook

Python: tweepy and facebook-sdk

other open-source tools exist

Integration with quanteda is fairly straightforward

Connecting through R

R packages

- Twitter: twitteR for REST, streamR for Streaming
- Facebook: Rfacebook

Python: tweepy and facebook-sdk

other open-source tools exist

Integration with quanteda is fairly straightforward

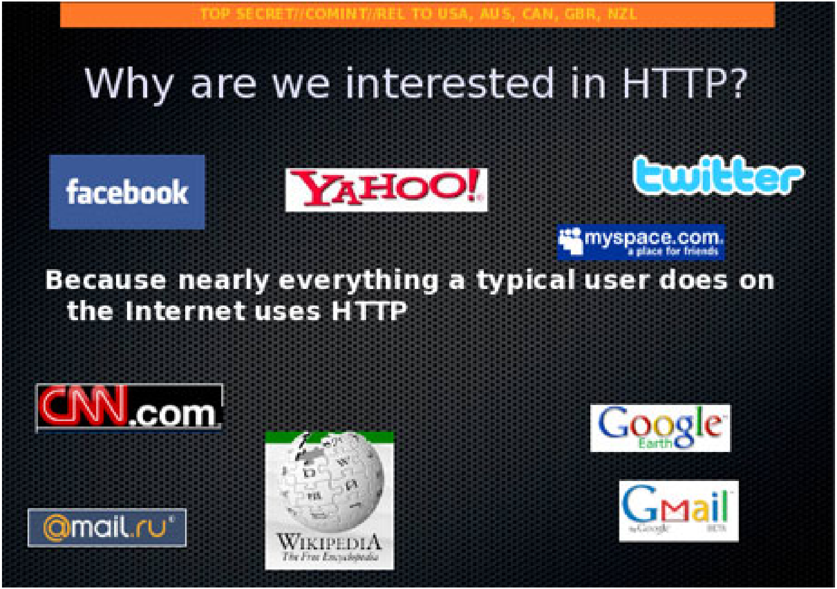
Other social media access packages

- `tumblR` R interface to the Tumblr web API
- `instaR` R interface to Instagram API
- `Rlinkedin` R interface to LinkedIn API
- `RedditExtractoR` R interface for Reddit API

How can we access this data?

- API: Application Programming Interface — a way for two pieces of software to talk to each other
- Twitter, facebook, google — all expose public web services
- Your software can receive (and also send) data automatically through these services
- Data is sent by `http` — the same way your browser does it
- Most services have helping code (known as a wrapper) to construct http requests
- both the wrapper and the service itself are called APIs
- http service also sometimes known as REST (REpresentational State Transfer)

HyperText Transfer Protocol



TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

Why are we interested in HTTP?

Because nearly everything a typical user does on the Internet uses HTTP

facebook
YAHOO!
twitter
myspace.com
a place for friends
CNN.com
Google
Earth
Gmail
by Google
@mail.ru
WIKIPEDIA
The Free Encyclopedia

Anatomy of a http request

```
https://api.twitter.com/1.1/search/tweets.json?
q=Nick+Clegg%21&since_id=24012619984051000&max_id=25012619
```

```
Nick Clegg! becomes Nick+Clegg%21
```

- Parameters to the API are encoded in the URL
- you must encode requests — spaces and non ASCII characters are replaced

Available social media APIs

- Wikipedia: mediawiki
- Google
- google plus
- blogger
- reddit
- foursquare
- facebook
- twitter: ‘Gardenhose’ (REST, Streaming), firehose, commercial

The twitter APIs: REST

- This is the most comprehensive API
- Returns a sample of historical data from the last 8–10 days.
- Stateless: you send a command and receive a result.
- http GET requests return information
- http POST requests upload or alter information (e.g. twitterbots)
- The manual: `https://dev.twitter.com/rest/public`
- R package : twitteR

The twitter APIs: Streaming

- Connect to the twitter server and collect tweets as they fly by.
- The manual:
  `https://dev.twitter.com/streaming/public`
- R package: streamR

Authentication

- Username and Password
- Oauth (ROauth): share a key without sharing a username and password
- IP address limitations
- Rate limitations
- Per-user and per-application

Other options

- The firehose: work with twitter
- Commercial options: GNIP (now bought by twitter) and Datasift

The Output: JSON and XML

- XML: eXtensible Markup Language: encodes documents in a form that is both human-readable and machine readable
- JSON : JavaScript Object Notation
- If you have a choice, you probably want JSON
- JSON uses key:value pairs, XML uses trees
- JSON is easily read into a programming language
- Sometimes known as serialization formats

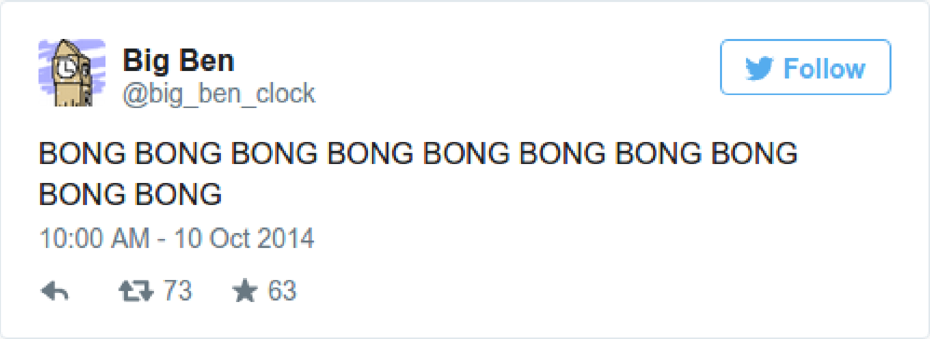
And finally... the data.

- Full of spam, bots, unicode, and gibberish
- Homographs and ambiguities are a problem, e.g. Clegg, Cameron, Miliband
- Lots of retweets (approximately one-third retweets, replies, tweets)
- Only 1% show location — some methods exist to infer location
- All aspects of metadata and reply/retweet structure are available
- All aspects of network structure: followers and ‘friends’, profile information

Twitterbots

- API also allows actions such as posting tweets (POST)
- Examples:
- @netflix_bot posts new content using netflix api
- @eqbot posts earthquake warnings
- @pentametron posts pairs of tweets in rhyming couplets [3]

Twitterbots



Big Ben
@big_ben_clock

BONG BONG BONG BONG BONG BONG BONG BONG BONG BONG

10:00 AM - 10 Oct 2014

How to get visible content directly from web pages

Scraping text from the web

- web crawlers/spider download sites by traversing links
- Python - scraPy, Beautiful Soup
- R - Rvest
- Chrome web plugins, import.io
- cUrl, wget, or other tools available ('httrack')
- Problems: rate limiting, ethical issues

Make scraping unnecessary!

- Organizations and governments should be aware of need for open, machine-readable data
- data.gov.uk, data.gov
- Data should be available in human and machine format!
- Make the raw data available in as many formats as possible.
- Consider machine readability at time of data collection
- Provide an Application Programming Interface (API)